

**EUROPEAN PATENT APPLICATION**

**(51) Int. Cl.<sup>6</sup>: G06F 13/40**

②② Date of filing : 09.06.95

**Inventor : Boury , Bechara**  
**3008 N.W. 28th Avenue**  
**Boca Raton, Florida 33434 (US)**

**Inventor : Frey, Bradly George**  
**1441 Braided Rope Drive**  
**Austin, Texas 78727 (US)**

**Inventor: Reid, James D.**

**600 Mission Hill Road  
Boynton Beach, Florida 33435 (US)**

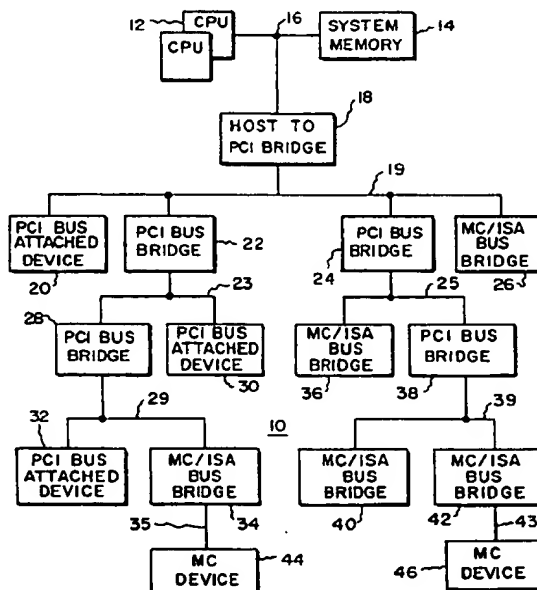
Inventor: Valli, Ronald

**2598 N.W. 36th Street  
Boca Raton, Florida 33434 (US)**

**(74) Representative : Davies, Simon Robert**  
**I B M**  
**UK Intellectual Property Department**  
**Hursley Park**  
**Winchester, Hampshire SO21 2JN (GB)**

**(54) Bus bridge**

(57) A computer system typically comprises a processor 12, a first bus 16 for conducting signals in accordance with a first bus protocol that does not support I/O address signals; a second bus 19 for conducting signals in accordance with a second bus protocol that supports input/output (I/O) address signals; and a bridge circuit 18 for coupling the first bus to the second bus. The processor includes a circuit for emitting address signals and an address type signal directed to a selected peripheral device. The bridge circuit comprises a filter 52 for determining whether the address signal emitted by the processor corresponds to a peripheral device coupled to a bus subordinate to the bridge circuit; and a translation circuit 54, coupled to the filter, for translating signals in accordance with the first bus protocol to signals in accordance with the second bus protocol for transmission to the selected peripheral device.



**FIG. 1**

The present invention relates to computer systems, and more particularly to computer systems having different or multiple bus architectures.

Computer systems typically include more than one bus, each bus in the system having devices attached thereto which communicate locally with each other over the bus. Examples of the different types of buses present in typical computer systems are a system bus to which a host central processing unit is attached and one or more peripheral buses. System-wide communication over different buses is required, however, if a device attached to one bus needs to read or write information to or from a device on another bus. To permit system-wide communication between devices on different buses, bus-to-bus bridges (interfaces) are provided to match the communications protocol of one bus with that of another.

Each bus-to-bus bridge in a multiple-bus computer system is used to connect two buses in the system. Various types of buses are available to construct a given computer system. One such bus which is becoming widely accepted is the PCI (Peripheral Component Interconnect) bus, which is capable of performing significant data transfer in a relatively short period of time. The PCI bus achieves this high level of performance, in part, because it may be directly linked to other high speed buses, such as system buses to which a CPU may be connected, and thus may provide for rapid transfer of data between devices attached to the PCI bus and devices attached to the system bus. In fact, the operation of several high integration devices, such as certain graphics package controllers, requires a direct link to a system bus through a high performance bus such as the PCI bus. In addition, the PCI bus architecture does not require any "glue logic" to operate peripheral devices connected to it. Glue logic for other buses typically consists of miscellaneous hardware components such as decoders, buffers or latches that are installed intermediate the peripheral devices and the bus.

The primary PCI bus operates on a synchronous clock signal of 33 MHz, and the strings of data transmitted over the PCI bus are 32 bits wide. A 32-bit data string on the PCI bus is called a double word (DWORD), which is divided into 4 bytes each comprising 8 bits of data. The address and data information carried by the PCI bus are multiplexed onto one signal. Multiplexing eliminates the need for separate address and data lines, which in turn, reduces the amount of signals required in a PCI bus environment as opposed to other bus architectures. The number of signals required in PCI bus architecture is between 45-47 while non-multiplexed buses typically require twice this number. Accordingly, because the number of signals are reduced, the number of connection pins required to support a device linked to the PCI bus is also reduced by a corresponding number. PCI architecture is thus particularly adapted for highly integrat-

ed desktop computer systems.

A more detailed description of the structure and operation of PCI bus architecture is provided in "Peripheral Component Interconnect (PCI) Revision 2.0 Specification", published April 30, 1993 and "Preliminary PCI System Design Guide", revision 0.6, published November 1, 1992 by the PCI Special Interest Group. It is assumed that the reader of this application is familiar with the contents of both these references.

Personal computers and workstations are required to have expansion capability which allows the user to add adapters and devices to the base system after initial installation. To satisfy this requirement, these systems contain expansion buses. The two most common are the Industry Standard Architecture (ISA) bus and the Micro Channel (MC) bus. To satisfy user requirements for higher reliability and performance (e.g., as in servers) personal computer systems are being built with more and faster processors, faster networks and redundancy. The processors include Intel x86, RISC and other microprocessors. Some of these microprocessors do not support I/O address space. In addition, redundancy requirements have driven system designers to include more expansion slots in multiple expansion channels (for example, IBM Model 295 servers have two Micro Channels). Future products will include multiple buses.

As processor and I/O device speeds increased, the industry direction has been to satisfy the demands for high speed I/O functions by implementing local buses. Local buses support higher speed devices by attaching them in close proximity to the processor.

Personal computer systems constructed with the PCI or other local bus supporting multiple Micro Channel buses introduce an environment that may contain multiple instances of the same adapter. Since many adapters have been designed to support only two, three, or four instances of the adapter in a system, addressing conflicts arise when more adapters are installed. There are processor and system architectures which do not provide instructions or signals to support I/O address space. In these systems, I/O devices must be mapped into the system's memory address space. There are two problems to be solved: (1) multiple instances of adapters with address selection limitations; (2) supporting adapters with I/O addresses in systems where processors do not support I/O address space.

New functions and features in personal computers have introduced a need for I/O addresses beyond the 64k supported by the Intel x86 architectures. Moreover, the address space architecture should support: Processors with and without I/O address space; Systems with multiple processors; Systems with hierarchical buses; Systems with multiple expansion buses (Micro Channel, ISA, or EISA); Addressability of system memory and non-system (I/O) mem-

ory by system master; Addressability of I/O devices by System masters; Addressability of system memory and non-system memory by bus masters on any bus; Addressability of I/O devices by bus masters on any bus; and Protection of I/O addresses.

To assure proper system operation, it is necessary to define the methods for generating and transmitting addresses throughout the system. These definitions impose requirements on the devices used to implement the bus bridges and the software that configures and uses the system.

In today's environment the address space architecture may be required to be compatible with the following architectures: (1) Micro Channel; (2) Industry Standard Architecture (ISA); (3) Peripheral Component Interface (PCI); (5) Personal System/2; (7) PowerPC; and (8) Personal Computer Memory Card International Association (PCMCIA).

There are two types of entities in a system that emit address and control signals to the buses: (1) System master; (2) Bus masters and third party DMA controllers. System masters manage and control the system configuration. The system master is typically the host processor, and there may be more than one processor in the host processor set. Bus masters arbitrate for control of the bus. Bus masters support data transfers with I/O and memory slaves. DMA (direct memory access) controllers do not arbitrate for the bus on behalf of a slave; a DMA slave arbitrates for the bus. DMA controllers support data transfers between DMA slaves and memory and I/O slaves. DMA controllers arbitrate for the bus for memory-to-memory transfers.

Each of these entities has addressing constraints imposed by preexisting architectures that must be accommodated. The system master controls the configuration of all of the system resources and is the only element in the system that has knowledge of the bus topology and the address transmission mechanisms.

The system master addressing constraints include: (1) System memory must be located at physical address 0; (2) Location FFFFFFF0h must contain initialization code; and (3) Processors without I/O instructions and signals must communicate with devices physically mapped in the I/O address space. Memory controllers and operating systems for RISC processors are not typically designed to accommodate 20 and 24 bit memory address range discontinuities. Intel x86 processors emit only 16 bit I/O addresses. Processors in multiprocessor environments have private local system memory. Not all of these constraints apply to all system masters. All system masters in a system must be the same type.

There are several constraints caused by the bus masters and DMA controllers. Bus masters and DMA controllers must emit address and control signals to facilitate the movement of data without intervention from the system master. The target of the address and

control signals is either a memory slave or an I/O slave. (Bus masters are not permitted access to Micro Channel or PCI configuration space.) Bus masters may be attached to the PCI bus or to the Micro Channel bus. The following architecture limitations impose constraints on the address space architecture: (1) Micro Channel and ISA architectures support only 16 bit I/O addresses; (2) Micro Channel and ISA adapters can contain I/O functions mapped in the memory address space which are limited to 20 bit and 24 bit addresses. Systems with multiple Micro Channel or ISA expansion buses may be configuration limited by device address range selection capability.

Accordingly, the invention provides a computer system comprising: processor means including means for emitting an address signal directed to a selected peripheral device; a first bus; a second bus; a bridge circuit for coupling the first bus to the second bus, the bridge circuit comprising: filter means for determining whether the address signal emitted by the processor corresponds to a peripheral device coupled to a bus subordinate to the bridge circuit; and translation means, coupled to the filter means, for translating an input address corresponding to the peripheral device selected by the processor into a signal for transmission to the selected peripheral device.

In a preferred embodiment, the first bus is directly coupled to the processor, generally the first bus being the host bus, and the second bus is an I/O bus. The translation means translates an input address corresponding to the peripheral device selected by the processor to an output address compatible with the peripheral device. The first bus supports n-bit address signals for selecting a peripheral device and the second bus supports m-bit I/O address signals for selecting the peripheral device, and the translation means translates an n-bit address signal into an m-bit address signal.

Typically to produce an output address for transmission to the selected peripheral device, the translation means may comprise an adder for adding a predetermined value to the input address, an exclusive OR device for performing an OR function on a predetermined value and the input address, or a stripping circuit for removing a predetermined number of high order bits from the input address.

Preferably, the filter means comprises multiple address range register sets, each register set comprising a start and end value, an input signal type, and an output signal type.

In one embodiment, the first bus conducts signals in accordance with a first bus protocol that does not support I/O address signals, said second bus conducts signals in accordance with a second bus protocol that supports input/output (I/O) address signals, and said translation means translates signals in accordance with the first bus protocol to signals in accordance with the second bus protocol.

From another aspect, the invention provides an information processing system comprising:

a processor including means for emitting address signals and an address type signal directed to a selected peripheral device;

a first bus for conducting signals in accordance with a first bus protocol that does not support I/O address signals;

a second bus for conducting signals in accordance with a second bus protocol that supports input/output (I/O) address signals; and

a bridge circuit for coupling the first bus to the second bus, the bridge circuit comprising:

filter means for determining whether the address signal emitted by the processor corresponds to a peripheral device coupled to a bus subordinate to the bridge circuit; and

translation means, coupled to the filter means, for translating signals in accordance with the first bus protocol to signals in accordance with the second bus protocol for transmission to the selected peripheral device.

From another aspect the invention provides an information processing system comprising:

a processor including means for emitting an address signal to a selected peripheral device;

an n-bit first bus coupled to the processor;

an m-bit second bus for coupling to input/output (I/O) and memory devices; and

a bridge circuit for coupling the first bus to the second bus, the bridge circuit comprising:

filter means for determining whether the address signal emitted by the processor corresponds to a peripheral device coupled to a bus subordinate to the bridge circuit; and

translation means, coupled to the filter means, for translating n-bit address signals to m-bit address signals.

From another aspect the invention provides an information processing system comprising:

a processor including means for emitting an address signal directed to a selected peripheral device and a memory map including a plurality addresses;

a first bus and a second bus; and

a bridge circuit for coupling the first bus to the second bus, the bridge circuit comprising:

filter means for determining whether the address signal emitted by the processor corresponds to a peripheral device coupled to a bus subordinate to the bridge circuit; and

translation means, coupled to the filter means, for translating an input address corresponding to the peripheral device selected by the processor to an output address compatible with the peripheral device.

The invention further provides a bridge circuit for coupling a first bus to a second bus, wherein the first bus complies with a first bus protocol that supports

I/O address signals and the second bus complies with a second bus protocol that does not support I/O address signals, the bridge circuit comprising:

filter means for determining memory and I/O ranges corresponding to the I/O or memory devices coupled below the bridge circuit; and

translation means for translating signals complying with the first protocol to signals complying with the second protocol for transmission to a selected peripheral device coupled to a bridge subordinate to the bridge circuit.

Thus, an information processing system may comprise a processor including means for emitting an address signal directed to a selected peripheral device; a first bus for conducting signals in accordance with a first bus protocol; a second bus for conducting signals in accordance with a first bus protocol; and a bridge circuit for coupling the first bus to the second bus. The bridge circuit comprises filter means for determining whether the address signal emitted by the processor corresponds to a peripheral device coupled to a bus subordinate to the bridge circuit; and translation means, coupled to the filter means, for translating signals in accordance with the first bus protocol to signals in accordance with a second bus protocol for transmission to the selected peripheral device.

FIG. 1 is a block diagram of a computer system;

FIG. 2 shows the bus-to-bus bridge address filter/translation component from Figure 1;

FIG. 3 shows an adder address translation apparatus;

FIG. 4 shows memory and I/O address spaces;

FIG. 5 shows the three regions of memory in an Intel x86 system;

FIG. 6 shows merged I/O map spaces;

FIG. 7 shows multiple buses mapped into memory and I/O address space;

FIG. 8 shows a 4GB segmented I/O address space;

FIG. 9 shows views of the system by the system master and bus master;

FIG. 10 shows a modification of FIG. 1 to include segment and bus bridge registers;

FIG. 11 shows extending the CPU 16 bit I/O addresses to 32 bits; and

FIG. 12 shows an address filtering and translation variable data field.

Referring to FIG. 1, a simplified block diagram of a computer system 10 is shown. Such a computer system may contain a PCI bus as the primary bus with Micro Channel or ISA buses subordinate to the primary PCI bus. The PCI bus structure is hierarchical and PCI buses may also be subordinate to the primary PCI bus.

In the system of Figure 1, one or more processor units 12 and a memory 14, both coupled to a host (or primary) bus 16, interface with an I/O bus 19 via a host-to-bus bridge 18. The host bus 16 conducts

processor-unique signals. The host-to-bus bridge 18 converts the processor-unique signals of bus 16 to standardized I/O signals for bus 19. In a preferred embodiment, the I/O bus 19 (which is subordinate to the bridge 18) and subordinate buses 23, 25, 29, and 39 are designed in compliance with the PCI standard, although the invention may be usefully employed in other I/O buses. The primary bus 16 is coupled to other PCI, Micro Channel (MC) or ISA buses via a hierarchical bus structure. The system 10 has the capability of supporting multiple expansion buses. The I/O bus 19 is coupled to a peripheral device 20, PCI bus bridges 22 and 24, and a MC/ISA bridge 26. PCI bus bridge 22 is coupled to another PCI bridge 28 and to a peripheral device 30, via PCI bus 23. The purpose of PCI bus-to-PCI bus bridges is to increase expandability (i.e., scalability). The PCI bridge 28 is coupled to a peripheral device 32 and a MC/ISA bus bridge 34, via bus 29. A Micro Channel device 44 is coupled to the bridge 34 via a Micro Channel bus 35. The PCI bus bridge 24 is coupled to a MC/ISA bridge 36 and to another PCI bus bridge 38 (via PCI bus 25). Bus bridge 38 is, in turn, coupled to a MC/ISA bus bridges 40 and 42, via bus 39. A Micro Channel device 46 (identical to device 44) is coupled to the bridge 42 via a Micro Channel bus 43. Bridges such as bridge 34 must convert the PCI-compliant signals of its primary bus 29 to the standardized signals required in its subordinate bus 35. The bus 35 may be a Micro Channel, ISA, EISA, or other standard bus architecture.

The bus hierarchy is compatible with both the PCI architecture and the Micro Channel architecture. In a preferred embodiment, the processor complex 12 includes one (or more) Power PC microprocessors which emit 32 bit addresses. Thus, in this structure, the processor complex 12 emits a 32-bit memory address to identify an I/O device.

Processor-to-host-bus bridges must be capable of translating memory addresses to I/O addresses on the primary PCI bus if the bridge is to be used in a RISC based system.

The address emitted must be properly directed through the bus structure to the correct destination, either an I/O device or a memory location. To meet the timing demands of the bus architectures, the bridge devices must be able to identify addresses of memory and I/O devices that are subordinate to (i.e., below) their bus. The bridge must recognize the address and respond on the bus with the Device Select signals indicating that the address has been accepted and a device will respond.

In systems without I/O signals, a bridge must have the ability to accept a memory address and transfer the address to the subordinate bus as an I/O address. It may also be required to translate the memory address to a different value to correctly identify the selected I/O device. In accordance with the invention several translation functions can be performed to

provide an I/O signal recognizable by the selected I/O device. A memory controller/host-to-bus bridge must accept address and memory/I/O signals from the processor and perform the following functions: (1) Address range filtering and routing; (2) Address translate.

Thus, PCI to PCI bridges (such as bridges 22 and 24) must provide address range filtering for both memory and I/O addresses. I/O address segments may be filtered using the address filter mechanism. Expansion bus bridges must accept address and memory/I/O signals (or commands) from the input bus and perform the following functions: (1) Address range filtering; and (2) Address translation.

Referring to FIG. 2, a simplified block diagram of the content of an expansion bus bridge 50 is shown. The bridge 50 couples a PCI bus 56 to a Micro Channel (or other standard I/O bus architecture) bus 58, and includes an address comparator/range filter 52 and a translator 54. The address comparator/range filter 52 monitors the address lines of the bus 56 to detect an address range for the subordinate devices of the bridge 50. The address comparator/range filter 52 filters out addresses not corresponding to the subordinate devices. The translator performs the required translation from the PCI-specific signals of bus 56 into an I/O signal recognizable by the selected I/O device (this would include the M/I/O signal indicating that the address is an I/O address).

Expansion bus bridges must be capable of examining an input address range and an input signal to determine if the address belongs to a device on the expansion bus. The expansion bus bridge 50 must maintain the protocol for transactions with addresses that are forwarded to devices on the expansion bus. For the PCI bus, the expansion bus bridge must activate the Device Select signal for all addresses within the ranges specified in the address range registers. The address range filtering mechanism 54 supports the I/O segment check required for segmented 4GB I/O address ranges.

The address translate functions include: mapping a range of the input addresses to an output range and changing a range of input addresses from one type to another (i.e. memory to I/O or I/O to memory). The expansion bus bridge must contain an address translate function that can translate input memory or I/O addresses within a range to an output address that is compatible with the devices on the expansion bus.

The expansion bus bridge translate function converts one address range to another and converts one signal type to another (memory to I/O, I/O to memory). PCI to Micro Channel bridges and PCI to ISA bridges are required to convert 32 bits input addresses to 16 bits addresses for I/O addresses emitted to the expansion bus. The input address may be a 32 bit I/O or memory address, the output required may be a 16 bit I/O address. In addition to the address trans-

late, the output signal may be translated from a Memory or I/O to an I/O or Memory signal.

Referring to FIG. 3, there is shown a translation apparatus 60. The translation performed by translator 54 can be accomplished in at least three ways. The three ways discussed herein are: (1) addition; (2) exclusive OR; and (3) stripping. In the first case stated above, the translation function is an adder. Thus, the n-bit wide register 62 contained within the address filter register set contains the high n bits of a value to be added to the input address to produce the output address. Thus, in the example of FIG. 3, a 32-bit adder 66 adds the contents of bridge translation register 62 to the input address 64 to produce an output address 68. The addition operation is unsigned and the overflow is discarded. This allows an input address to be translated to an output address anywhere in the address range. Referring back to FIG. 1, assume that the processor 12 outputs an address signal corresponding to Micro Channel device 44 and that the I/O address of device 44 is 3F7h (i.e., device 44 responds to 3F7h). Each bridge (18, 22, 28 and 34) must determine that the address signal is addressed to a device attached to one of its subordinate buses. The bridge 34 must also translate the address on bus 29 to 3F7h. This can be done with the adder 60 of FIG. 3 as follows. The address (i.e., input address 64) to be output by CPU 12 corresponding to the selected device 44 is selected by determining which input address 64 when added to the value in register 62 results in the output address 3F7h. That input address value is then inserted into the memory map of CPU 12.

Referring to FIG. 7, the center memory map represents the view of the processor 12 of the addresses for the I/O devices. The CPU 12 emits a 32-bit address capable of distinguishing among adapters having the same I/O address but attached to different buses. Thus, for example, if adapter device 44 was mapped by configuration software at memory address 200003F7h, the CPU would emit that address for selecting the device 44. However, the device 44 is expecting an I/O address 3F7h. As the CPU output goes through the bus bridges it is translated to address 3F7h because the adapter 44 expects to see I/O address 3F7h.

A second translation approach is to substitute an exclusive OR function for the translation function block 66 shown in FIG. 3. An n-bit wide register contains the high n bits of a value to be exclusive ORed with the input address to produce the output address. The resulting address is output to the subordinate bus with the appropriate control signal.

A third translation operation would be to perform a stripping function with the block 66. The stripping translation removes the high order n bits from the address bus prior to the bus bridge propagating the address to the subordinate bus.

## ADDRESS SPACE REFERENCE MODELS

Referring to FIG. 4, an address space architecture may accommodate the following address spaces:

- (1) Intel x86 4GB three-region memory address space;
- (2) Linear 4GB memory address space (no I/O space);
- (3) Intel x86 64K I/O address space; and
- (4) PCI 4GB I/O address space.

These address spaces are accommodated in four models of the address space architecture:

- (1) Model 1: 4GB memory address space with 64K I/O address space;
- (2) Model 2: 4GB memory address space with a single 64K I/O address space for multiple buses;
- (3) Model 3: A single 4GB address space containing both memory and I/O addresses; and
- (4) Model 4: 4GB memory address space with 4GB I/O address space.

### MODEL 1: 4GB MEMORY ADDRESS SPACE, 64K I/O ADDRESS SPACE

In this model, a single 64K I/O address space is available, in addition to the memory address space. The 4GB memory address space is separated into 3 regions to maintain compatibility with systems that used early Intel x86 microprocessors. These regions are:

- (1) Region 1 - 0 to 1MB;
- (2) Region 2 - 1MB to 16MB; and
- (3) Region 3 - 16MB to 4GB.

The Intel x86 memory map is illustrated in FIG. 5. When system memory is present in a region, it starts at the beginning of the address range. System memory is always contiguous within a memory region. If non-system memory is present, it is allocated from the highest addresses of the address range to lower addresses. Non-system memory within a region is not required to be contiguous.

### MODEL 2: 4GB MEMORY ADDRESS SPACE, SINGLE 64K I/O SPACE, MULTIPLE BUSES

This address space model is the same as the model described in, Model 1 (4GB Memory Address Space, 64K I/O Address Space). In addition to a single expansion bus, this model supports multiple expansion buses (as shown in FIG. 1). Multiple Micro Channel expansion buses may be configured to share a single 64K I/O address space. The bus bridges determine which I/O commands are directed to devices on their bus. Type 1 variable data fields are used to identify the address ranges that the bridge must pass through to attached adapters. Figure 6 illustrates the merging of two expansion bus I/O ad-

dress spaces. There is no specific limit on the number of expansion buses that may be merged into a single I/O space. The limitations are imposed by physical and electrical characteristics of the systems.

### MODEL 3: 4GB ADDRESS SPACE CONTAINING MEMORY AND I/O

In systems with processor complexes that have only a memory address space (as that discussed in reference to FIG.1), I/O addressed devices must be mapped into a subset of the memory address space. The single address space model may also be used to avoid I/O address conflicts or expand the number of I/O addresses beyond the 64K limit.

As discussed above, mapping I/O addresses into the memory address space requires that the host-to-bus bridge recognize the memory address range that the I/O is mapped into and emit an I/O output signal to the bus for I/O devices attached to the primary PCI bus 19. It may also be necessary to translate the memory address to a different value before the bus bridge emits an I/O address to the selected device. I/O addresses are limited to 16 bit addresses only on expansion buses that are limited to 16 bit addresses.

Referring to FIG. 7, two MC buses mapped into memory and I/O address space are shown. FIG. 7 illustrates a system with two Micro Channels (MC1 and MC2) with the I/O and regions 1 and 2 non-system memory mapped into memory address space above 16M. The memory map shown in the center of the figure represents the CPU's view of where in the memory address space the devices attached to the MC1 and MC2 buses are located.

### MODEL 4: 4GB MEMORY ADDRESS, 4GB I/O ADDRESS SPACE

The PCI specification supports a 32 bit, 4GB I/O address space PCI devices are required to decode all 32 bits of the address bus for I/O bus commands. The Intel x86 processors do not, however, emit 32 bit I/O addresses.

The 4GB I/O address space may be used in the subset mode, utilizing only the first 64K of the I/O address space as described above under the caption "4GB Memory Address Space, 64K I/O Address Space." Merged 64K I/O addresses, as described in "4GB Memory Address Space, single 64K I/O Space, Multiple Buses" are also supported. PCI devices must be mapped into this 64K address space or mapped into the memory address space.

When used as a 4GB I/O address space, the I/O addresses may be emitted from the system master (e.g., processor 12 of FIG. 1) as: (1) 32 bit memory addresses; and (2) 16 bit I/O addresses.

The 32 bit memory address mode is described above under the caption: "4GB Address Space Con-

taining Memory and I/O". This mode is used in systems with processors that do not emit I/O signals. A region of the system memory address space is allocated for I/O addresses. Memory addresses emitted by the processor which are destined for I/O slaves must have the memory signal translated to an I/O signal. The incoming memory address may be translated to a different value before output. This is accomplished with address range check and translation registers in the host-to-bus bridge.

PCI bridges must identify which I/O address ranges are used by devices or buses attached to their bridge. All devices and expansion bus bridges must be programmed with addresses within a range (usually sequential). No other bridge within the system may have addresses within the same range. Expansion bus bridges must also identify which I/O addresses are below the bridge.

In the 4GB I/O address mode, one way of implementing a segmented I/O address map would be to consider the I/O address space to be divided into 64K segments of I/O address space, each 64K long as illustrated in Figure 8. In this mode, the processor emits a 16 bit I/O address with an I/O signal. An external I/O segment register is appended to the high order address bits of the address bus and the resulting 32 bit address is presented to the host-to-PCI bus bridge. Each PCI bus in the hierarchy is assigned a segment number and all devices on that bus have addresses with the high 16 bits equal to the segment value. Each Micro Channel expansion bus is assigned a segment value which is used to identify I/O addresses destined for devices on that bus. This addressing mode allows full protection of I/O addresses in Intel x86 based systems by controlling the setting of the segment register at ring 0 and using bit map permission to restrict device drivers in user space from accessing I/O resources they do not control.

A subset of this architecture allows an 8-bit implementation using the high order 8 bits of the address bus. This model of the I/O address space contains 256 64K I/O address segments which may be allocated to expansion buses. Using the high order address bus bits permits a wider distribution of the I/O addresses which may be necessary if the expansion buses are attached to different PCI buses.

### ADDRESS SPACE DOMAINS

Applying the requirements and constraints to the address reference model implies that multiple entities in a system must be assigned the same physical address. Assigning the same address to two physical entities creates unacceptable electrical conflicts. To resolve these conflicts, these entities are placed in separate address domains and address translation mechanisms must be provided at the domain boundaries. Specifically, address translations are provid-

ed at the bridges and the memory controller. This results in two views of the system address space: (1) The system master view; and (2) The bus master view.

The system master sees the address space before any translations are applied, typically with system memory starting at memory address 0. This is described in more detail in "System Master View of Address Space."

Bus masters see the address space after the translations are applied. It is possible for the bus master view of the memory address of the start of system memory to be a value other than 0.

Referring to FIG. 9, the system of FIG. 1 is modified (and simplified) to illustrate these two views of the address space. Note that system memory has the same relationship to the system master as I/O or non-system memory. This is a logical view of the structure and does not represent the electrical or bus structure.

The system master views the address space from above the host-to-bus bridge prior to any address translation. Typically, the system master sees system memory starting at address 0. The system memory may be contiguous or may contain compatibility holes. Non-system memory on one or more of the buses may be mapped into these holes. Memory space above the end of system memory may be occupied by memory mapped or I/O mapped I/O slaves.

The bus master view of the address space is shown in FIG. 9 under the dotted line. Translations may be applied at each bridge in the system. The following describes the address space map from the viewpoint of a bus master residing on any of the attached buses. This view considers the bus master as it addresses non-system memory or I/O slaves in using the different addressing models. There are two types of addressing transactions: (1) transactions between masters and slaves on the same bus only; (2) transactions between masters and slaves on different buses. In addition, transactions between masters on any bus and system memory must always be allowed.

A bus master addressing a slave on the same bus (Micro Channel or PCI) must emit an address to which the slave will respond and which the bridge will recognize belongs on the bus. The bridge above the master must recognize that an address emitted by a master is local to the bus and does not require negotiating for the bus above the bridge.

Bus masters on one expansion bus may emit addresses that are associated with devices on a different expansion bus. These addresses may be one of the following: 32 bit memory addresses; 24 bit memory addresses; or 16 bit I/O addresses.

There are constraints associated with the 24 bit memory addresses and the 16-bit I/O addresses. In systems with multiple expansion buses and a single shared memory region 2, the expansion bus bridge

must use the address range filter registers to detect if the address is local to the bus or remote (on another expansion bus). If the address is on another expansion bus, the bridge must gain control of the above bus and emit the address and appropriate signal upward.

Likewise, 32-bit memory addresses emitted by bus masters require the expansion bus bridge use the address range filters to determine if the address is below the bus or above the bus.

## ARCHITECTURE OF ADDRESS SPACES

This section describes the architecture of memory address spaces and I/O address spaces. The requirements are defined for: (1) Host-to-PCI bus bridges; (2) Micro Channel, ISA and EISA bus bridges; and (3) PCI-to-PCI Bus bridges.

An architecture in accordance with the invention could allow the 32-bit memory address space of Intel x86 and RISC processors to be mapped in one of two ways: (1) with Intel 20, 24 bit address range holes (see 3 region model of FIG. 4); or (2) with all system memory contiguous. The 32-bit memory address space contains two types of memory: (1) memory in the processor memory address space for operating system and application software and data (called system memory in the PC architecture); (2) Memory in the processor memory address space used by I/O devices (called non-system memory in the PC architecture).

There are two types of processors to consider, processors with and without I/O address space. These processors can emit some or all of the following: (1) 32 bit addresses; (2) 24 bit addresses; (3) 16 bit addresses; and (4) M/I/O signal.

Memory addresses emitted by a system master and used by I/O slaves may be either memory mapped I/O devices or I/O mapped I/O devices. I/O addresses that are mapped into memory addresses are always mapped into non-system memory. When a memory address is emitted from an Intel x86 processor the memory/I/O signal is in the memory state. All system memory addresses must be present in the memory address space at all times. Defective memory may be mapped out via page management mechanisms, however, defective memory is not considered non-system memory.

Memory addresses may be translated at certain interfaces (bus bridges). Therefore, a specific location of physical memory may appear to have different addresses as viewed by the system master and a master on a bus.

An architecture in accordance with the invention, permits the I/O address space to be mapped in the following modes: (1) one 64K I/O address space; (2) a 4GB I/O address space which may be segmented into 64K segments; or (3) I/O addresses mapped into the

memory address space. Each of these options is necessary to support one or more of the hardware configurations. Processors which do not support I/O addresses and the I/O signal require all I/O to be mapped in memory.

An I/O address emitted by an Intel x86 processor is a 16 bit address with the Memory/I/O signal in the IO state. Special instructions, IN and OUT are used in the Intel x86 processor to cause the I/O signal to be active. RISC processors typically do not emit I/O addresses or the I/O signal.

Existing I/O devices on adapters (both Micro Channel and ISA) decode I/O addresses in the 64K I/O address range and ignore the high order address bits above A15. These adapters typically have a limited set of I/O address selections. The number of I/O address selections an adapter supports is usually determined by the number of adapters expected to be installed in a system with a single expansion bus.

The PCI specification defines I/O cycles on the PCI bus with an I/O command and with a 32 bit address. PCI devices are required to decode all 32 bits of an I/O address. The I/O address space is shared by multiple devices, each using a relatively small number of I/O addresses (i.e. 1 to 32 addresses) which may not be contiguous. The architecture does not restrict the number of I/O addresses a device may use. The Micro Channel architecture allows adapters to consume larger amounts of I/O addresses if the I/O addresses are fully programmable.

Referring again to FIG. 1, the memory controller/host-to-bus bridge 18 performs an address range check on all memory addresses emitted by the processor(s) 12. The address is then routed either to the system memory 14 or to the bus 19. Memory addresses which are routed to the bus 19 may be emitted to the bus 19 as either bus memory cycles or bus I/O cycles. A translate output signal indicator is used to determine which type of cycle is emitted. All I/O addresses emitted by the processor 12 are routed to the bus 19 (except those I/O addresses used by the memory controller/host-to-bus bridge).

The address translation functions include: (1) Mapping a range of the input addresses to an output range; (2) Changing a range of input addresses from one type to another (i.e. memory to I/O or I/O to memory); (3) Expanding an address range (mapping 32 bytes to 41 bytes). In systems that require support for more than 64K of I/O address space, the processor complex must contain segment registers for the high order n bits of address to be emitted to the bus. There must be one segment register for each processor in the system master processor set. The segment registers must be memory mapped in a 4KB address region that does not contain any device I/O or processor critical functions to allow the processor access to the register at all times. These registers will contain n bits of addressing information which will be added to the

16 bit I/O address emitted by the system master. These segment register may be programmed to direct the I/O operation to the appropriate expansion bus or PCI device. In addition, to the address translate, the output signal may be translated from a Memory or I/O to an I/O or Memory signal.

## ADDRESS SPACE ARCHITECTURE INTERFACES

Implementing the above allows compatibility with both Intel x86 and RISC processor architectures. The embodiment previously discussed with reference to FIG. 1 included a processor that does not emit I/O signals. Referring now to FIG. 10, the system of FIG. 1 has been modified to include segment registers 13 associated with the system processor(s). In this embodiment the processor(s) 12 are of the type that emit 16 bit I/O signals and a M/I/O signal indicating that this is an I/O address (i.e., processors of the Intel x86 type). Intel x86 processors support only 16 bits of I/O address space. To extend this I/O memory space to 32 bits, an external segment register 13 is used for each processor 12. The segment register 13 contains the high order 16 bits of a 32-bit address. When the processor 12 issues an I/O command (signalled by the M/I/O output being in the I/O state), the segment register is strobed onto the high order bits of the address bus.

Referring to FIG. 11, an I/O segment register is shown for appending bits 16-31 to the low order 16 bits (11) emitted by the CPU 12. Thus, a 32-bit address signal is provided to the address component of the processor bus 16.

The architecture interfaces include: bus bridge interfaces and processor/central electronics complex (CEC) interfaces. The CEC is the circuitry generally connected to the CPU. Processor/CEC subsystems must contain I/O segment registers to augment the I/O address output of Intel x86 processors when used in a multiple bus environment where more than 16 bits are required for I/O addresses.

Referring to FIG. 12, bus bridges must contain n sets of programmable functions as follows: (1) Enable; (2) Input Signal; (3) Start Address; (4) End Address; (5) Output Signal; (6) Address translate; (7) Route; and (8) I/O Segment Registers. Address inputs of the designated Input Signal type (M/I/O) are range-checked to be within the Start and End Address. If the input address is within the specified range and the input signal is the type specified, the output address and output signal are emitted to the bus below. Each set of address filters may be enabled separately.

Before output, the address may be translated by the address translate field. The address translate function may be enabled or disabled. In the general case, the address translation field is an n-bit wide field that represents a binary value used to translate

an incoming address to an output address. Allowable address translate functions include: (1) Adder; (2) Exclusive OR; or (3) Stripping.

The adder translation function consists of a register containing an unsigned integer to be added to the upper n-bits of the input address. Overflow from this addition is discarded. This allows wrap around of the address space so that an address range can be mapped into a lower address range. The address translate may be used to convert a 16 bit or 24 bit address to a 32 bit address. The address translation may also be used to convert a 32 bit address to a 16 bit or 24 bit address. This translate function is described in Figure 10 on page 18. The exclusive OR translate function consists of a register containing n bits of binary information to be exclusive ORed with the high order n bits of the input address.

The stripping function consists of a register which identifies the number of high order bits of the 32 bit address to be discarded (or stripped) from the address before the address is forwarded to the bus. For the host-to-bus bridge, the address is routed as directed by the Route indicator (either to the system memory or to the bus).

Therefore, an address space is defined for information processing systems with one or more expansion buses and one or more processors. A hierarchical bus structure is supported wherein different buses are in compliance with different bus protocols. This allows common bus bridges for Intel x86 and RISC based computer systems, and supports a broad range of system designs.

## Claims

1. A computer system comprising:
  - processor means (12) including means for emitting an address signal directed to a selected peripheral device;
  - a first bus (16);
  - a second bus (19);
  - a bridge circuit (18) for coupling the first bus to the second bus, the bridge circuit comprising:
    - filter means (52) for determining whether the address signal emitted by the processor corresponds to a peripheral device coupled to a bus subordinate to the bridge circuit; and
    - translation means (54), coupled to the filter means, for translating an input address corresponding to the peripheral device selected by the processor into a signal for transmission to the selected peripheral device.
2. The computer system of claim 1, wherein the first bus is directly coupled to the processor.
3. The computer system of claim 2, wherein said first bus is a host bus.
4. The computer system of any preceding claim, wherein the second bus is an I/O bus.
5. The computer system of any preceding claim, wherein the translation means translates an input address corresponding to the peripheral device selected by the processor to an output address compatible with the peripheral device.
6. The computer system of claim 5, wherein the first bus supports n-bit address signals for selecting a peripheral device and the second bus supports m-bit I/O address signals for selecting the peripheral device, and wherein the translation means translates an n-bit address signal into an m-bit address signal.
7. The computer system of claim 6, wherein n is greater than m.
8. The computer system of claim 6, wherein n is less than m.
9. The computer system of claim 6, wherein n is equal to m and wherein the translation means comprises means for mapping a range of input addresses to an output range.
10. The computer system of any preceding claim, wherein the translation means comprises an adder for adding a predetermined value to the input address to produce an output address for transmission to the selected peripheral device.
11. The computer system of any of claims 1 to 10, wherein the translation means comprises an exclusive OR device for performing an OR function on a predetermined value and the input address to produce an output address for transmission to the selected peripheral device.
12. The computer system of any of claims 1 to 10, wherein the translation means comprises a stripping circuit for removing a predetermined number of high order bits from the input address to produce an output address for transmission to the selected peripheral device.
13. The computer system of any preceding claim, wherein the filter means comprises multiple address range register sets, each register set comprising a start and end value, an input signal type, and an output signal type.
14. The computer system of any preceding claim,

wherein said first bus conducts signals in accordance with a first bus protocol that does not support I/O address signals, said second bus conducts signals in accordance with a second bus protocol that supports input/output (I/O) address signals, and said translation means translates signals in accordance with the first bus protocol to signals in accordance with the second bus protocol.

5

10

15. The computer system of any of claims 1 to 13, wherein said first bus conducts signals in accordance with a first bus protocol that supports I/O address signals, said second bus conducts signals in accordance with a second bus protocol that does not support input/output (I/O) address signals, and said translation means translates signals in accordance with the first bus protocol to signals in accordance with the second bus protocol.

15

20

16. The computer system of any preceding claim, wherein the processor emits an address type signal with said address signal.

25

17. The computer system of any preceding claim, wherein the processor includes a memory map including a plurality of addresses.

18. The computer system of any preceding claim, wherein the filter circuit includes means for determining memory and I/O ranges corresponding to the I/O or memory devices coupled to a bus subordinate to the bridge circuit.

30

35

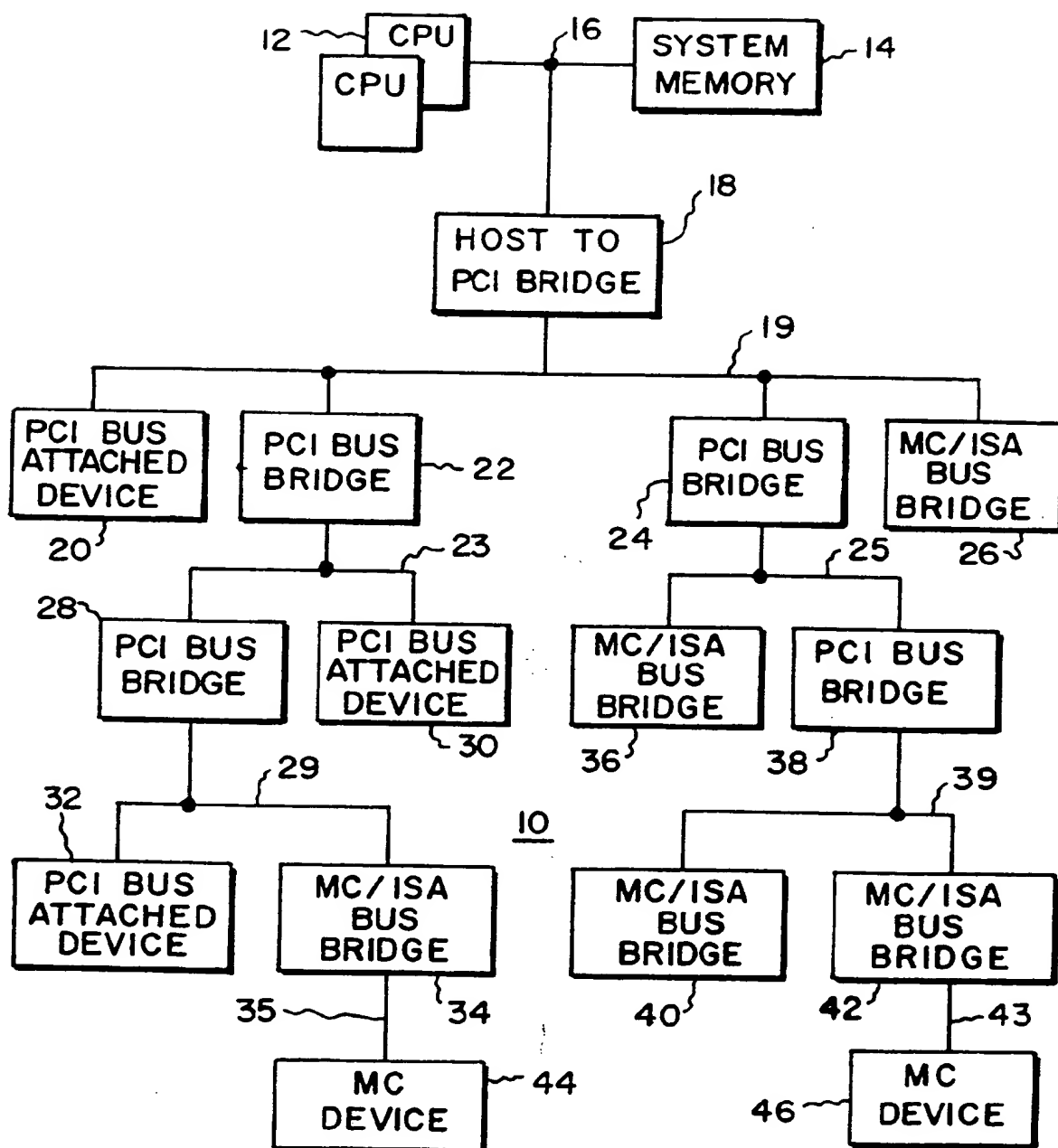
19. A bridge circuit suitable for use in the computer system of any preceding claim.

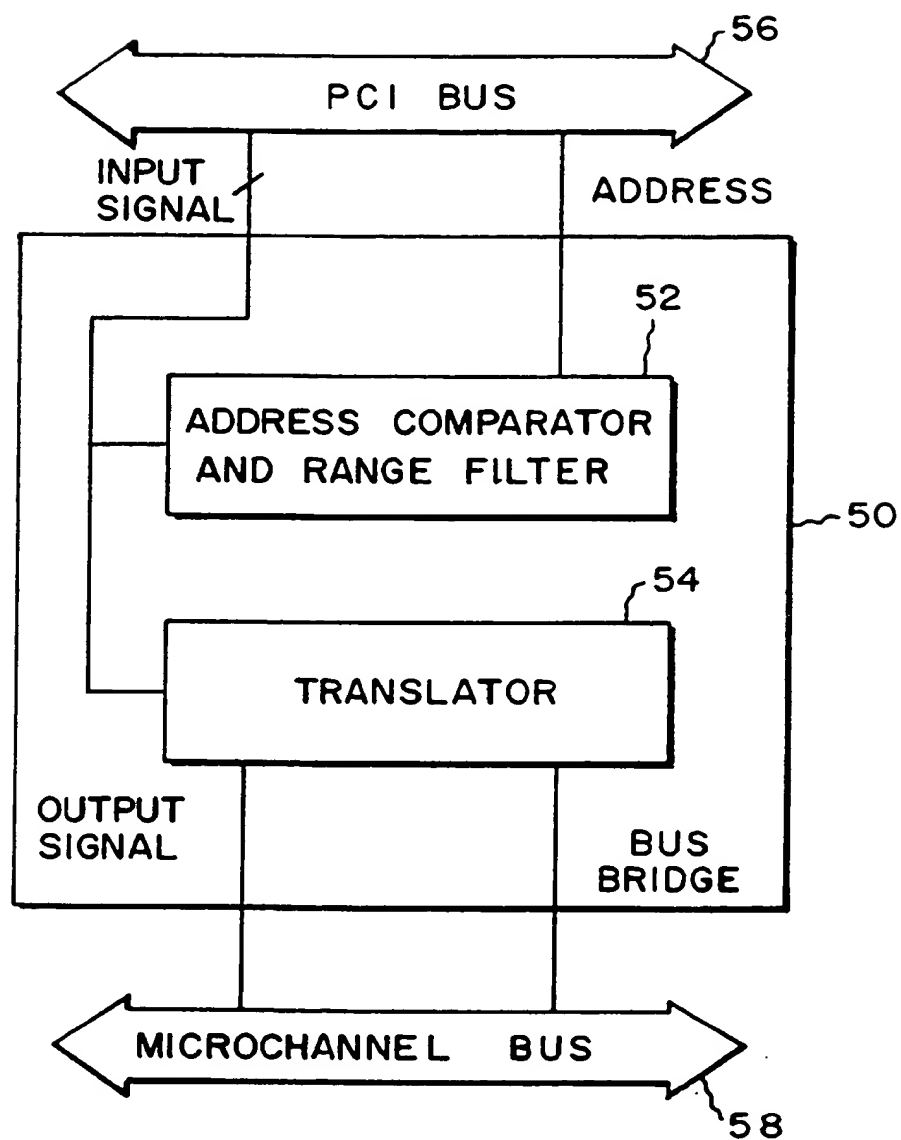
40

45

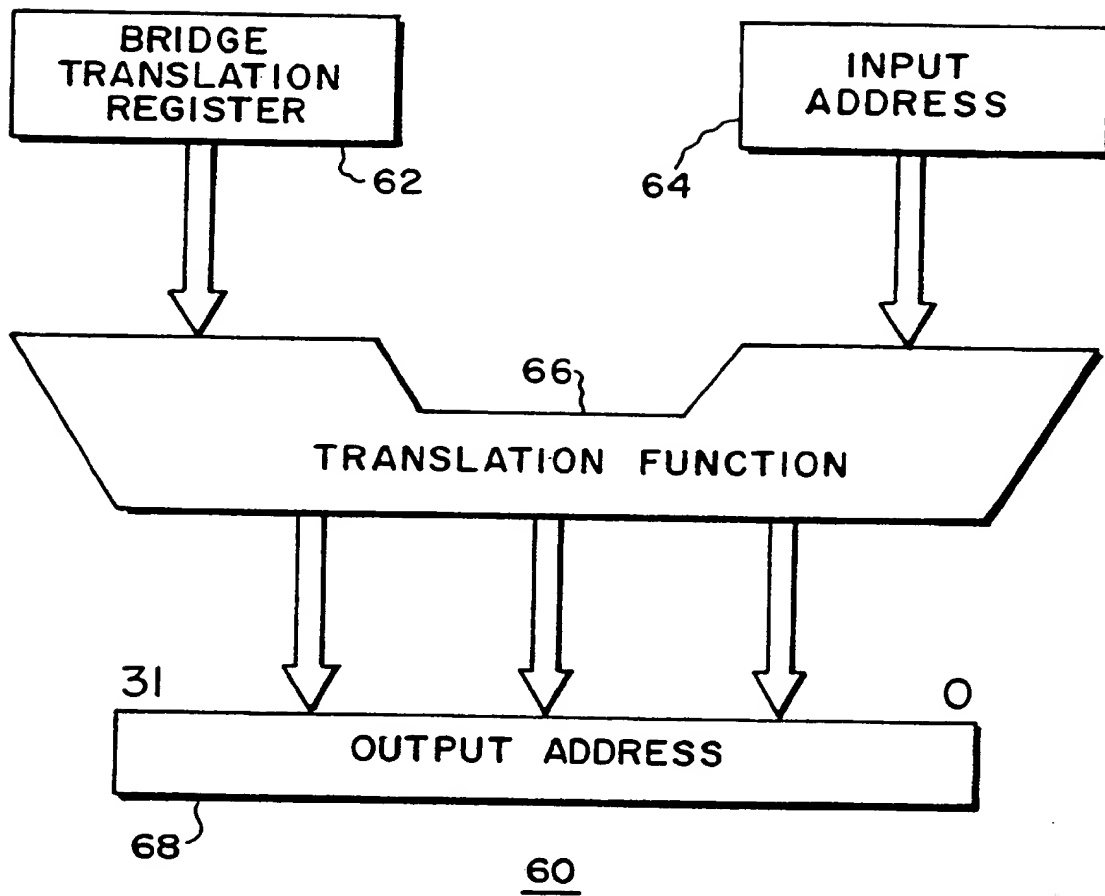
50

55

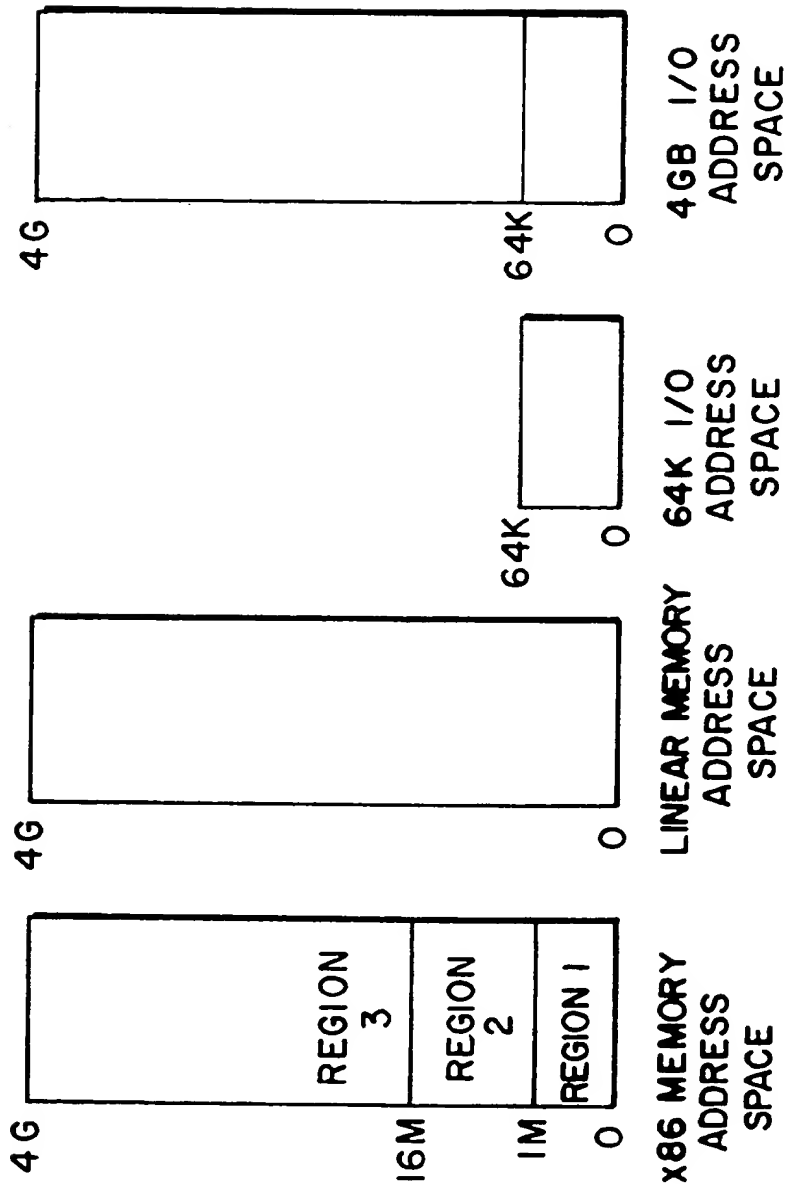
**FIG. 1**



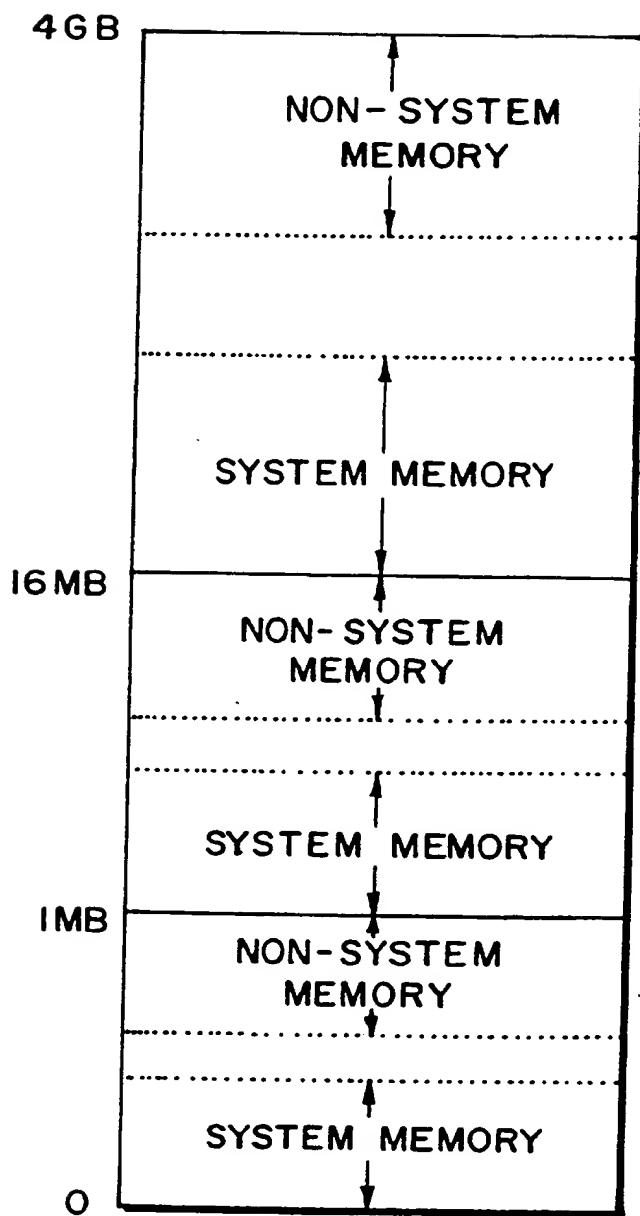
***FIG. 2***



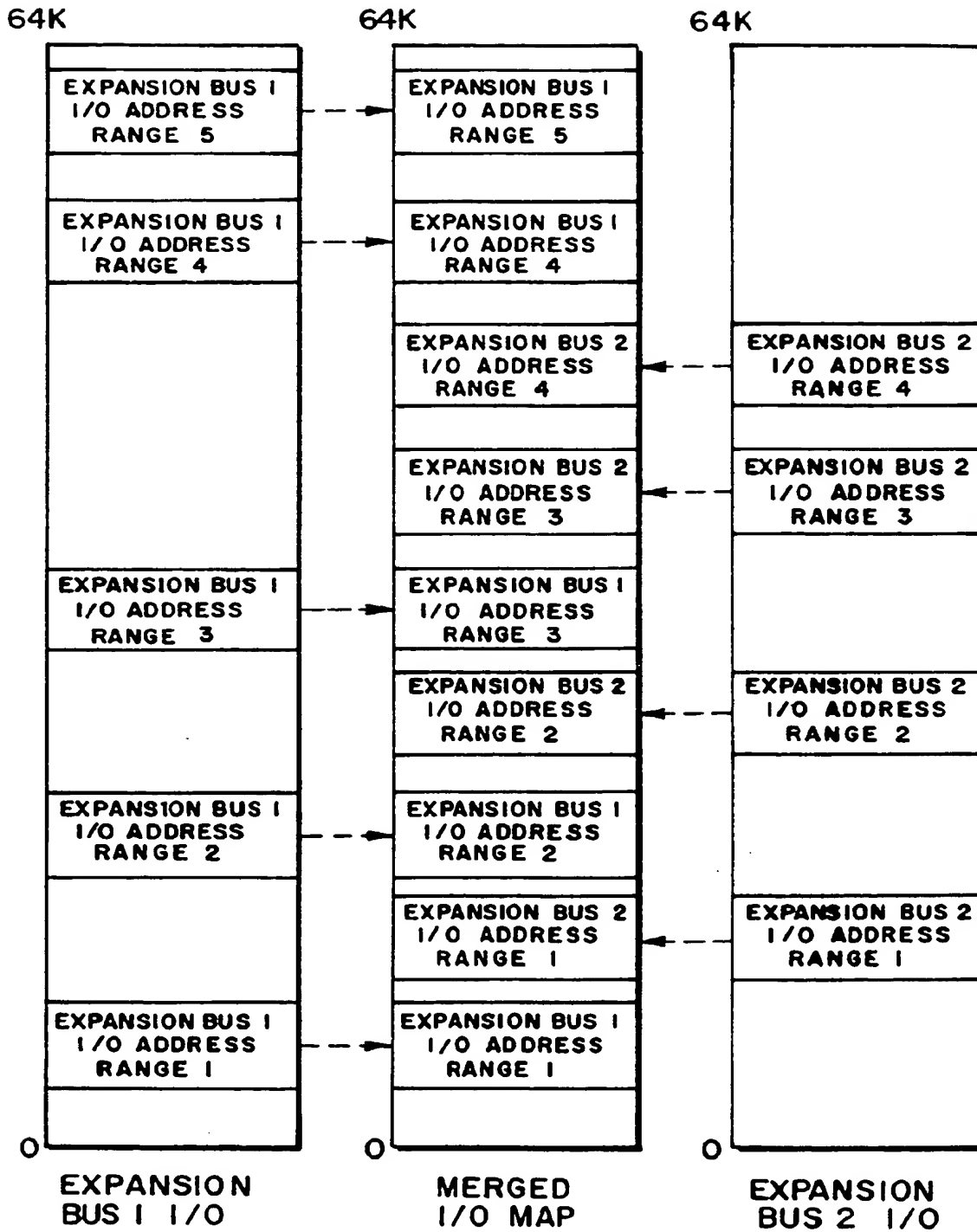
**FIG. 3**



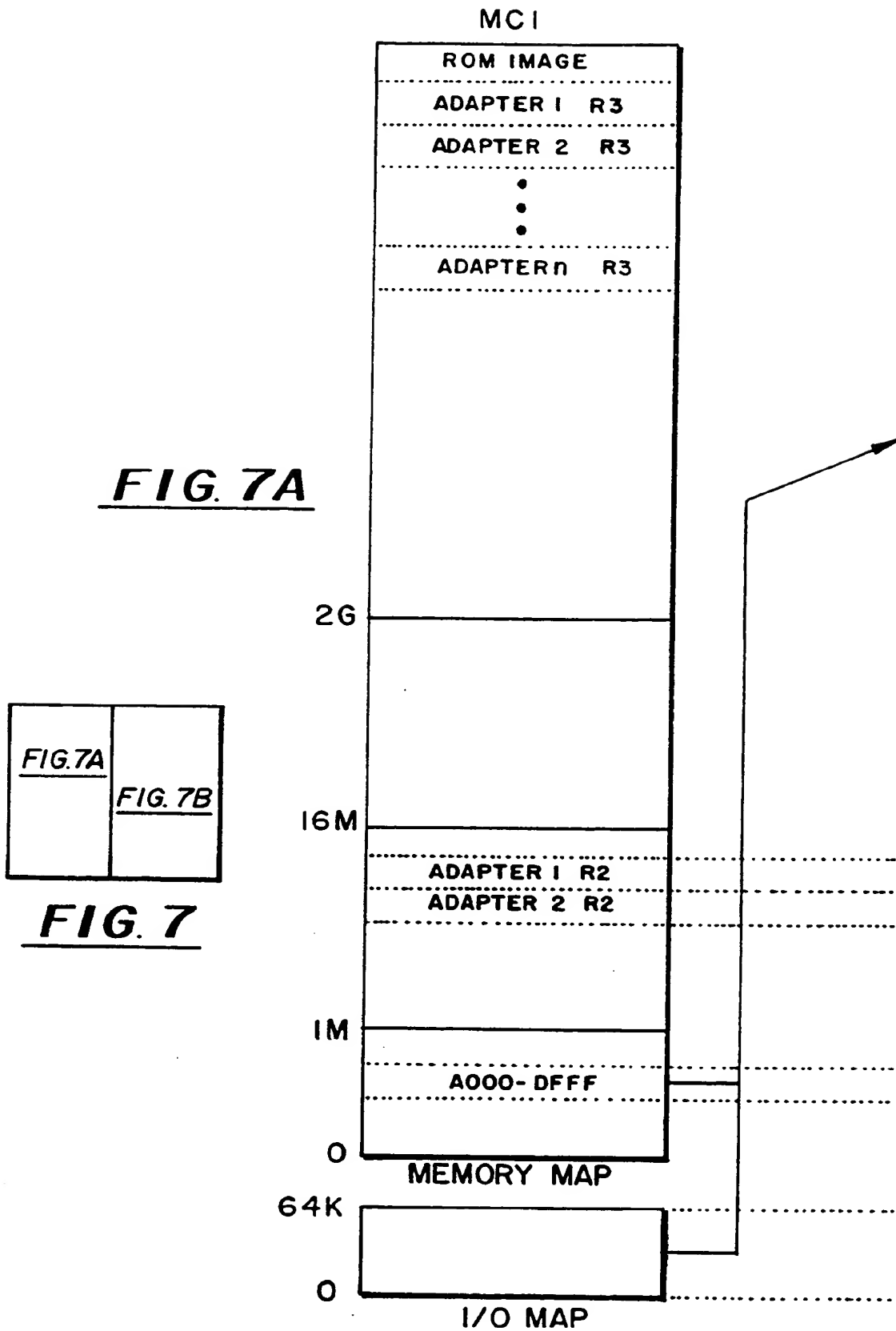
**FIG. 4**

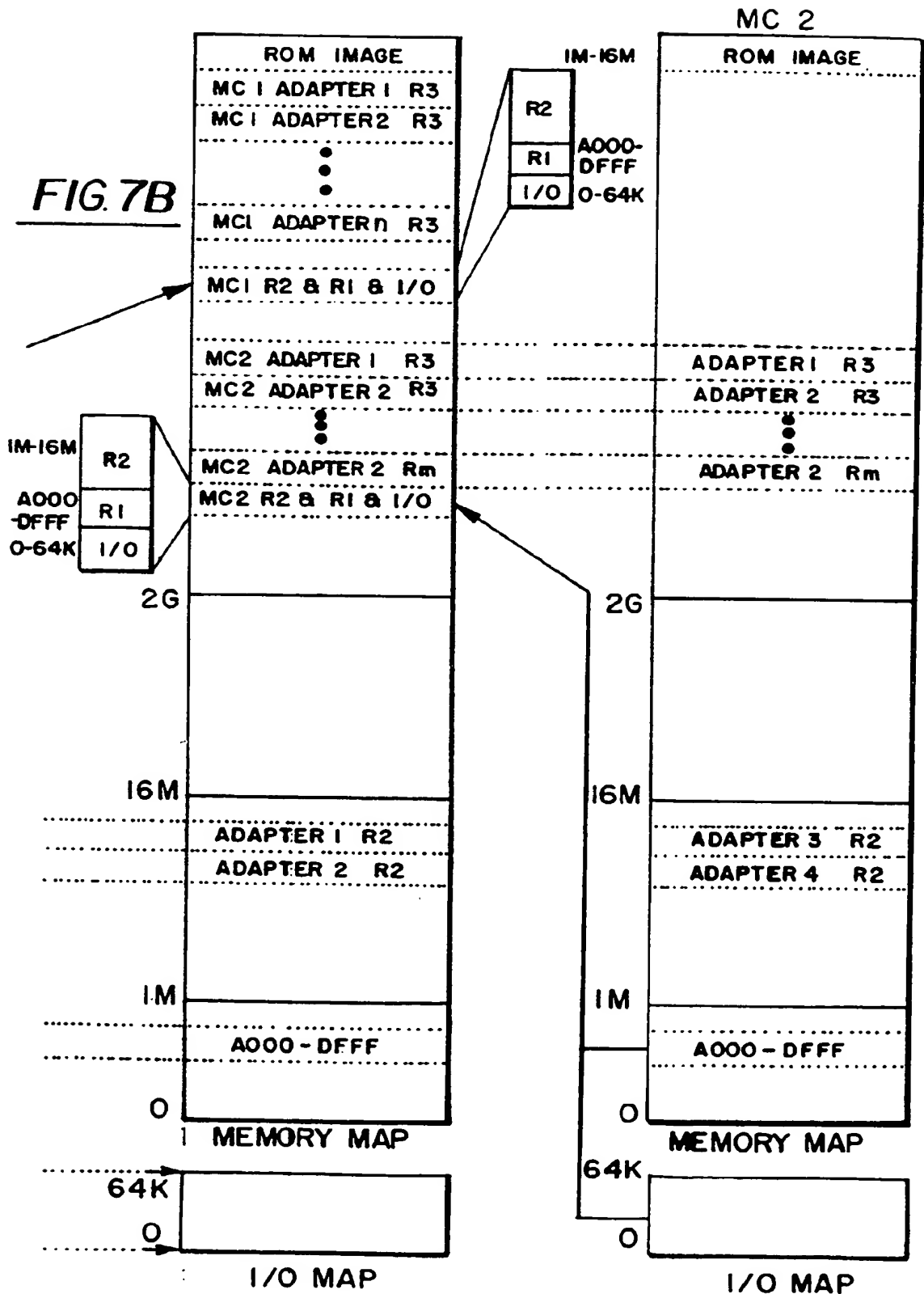


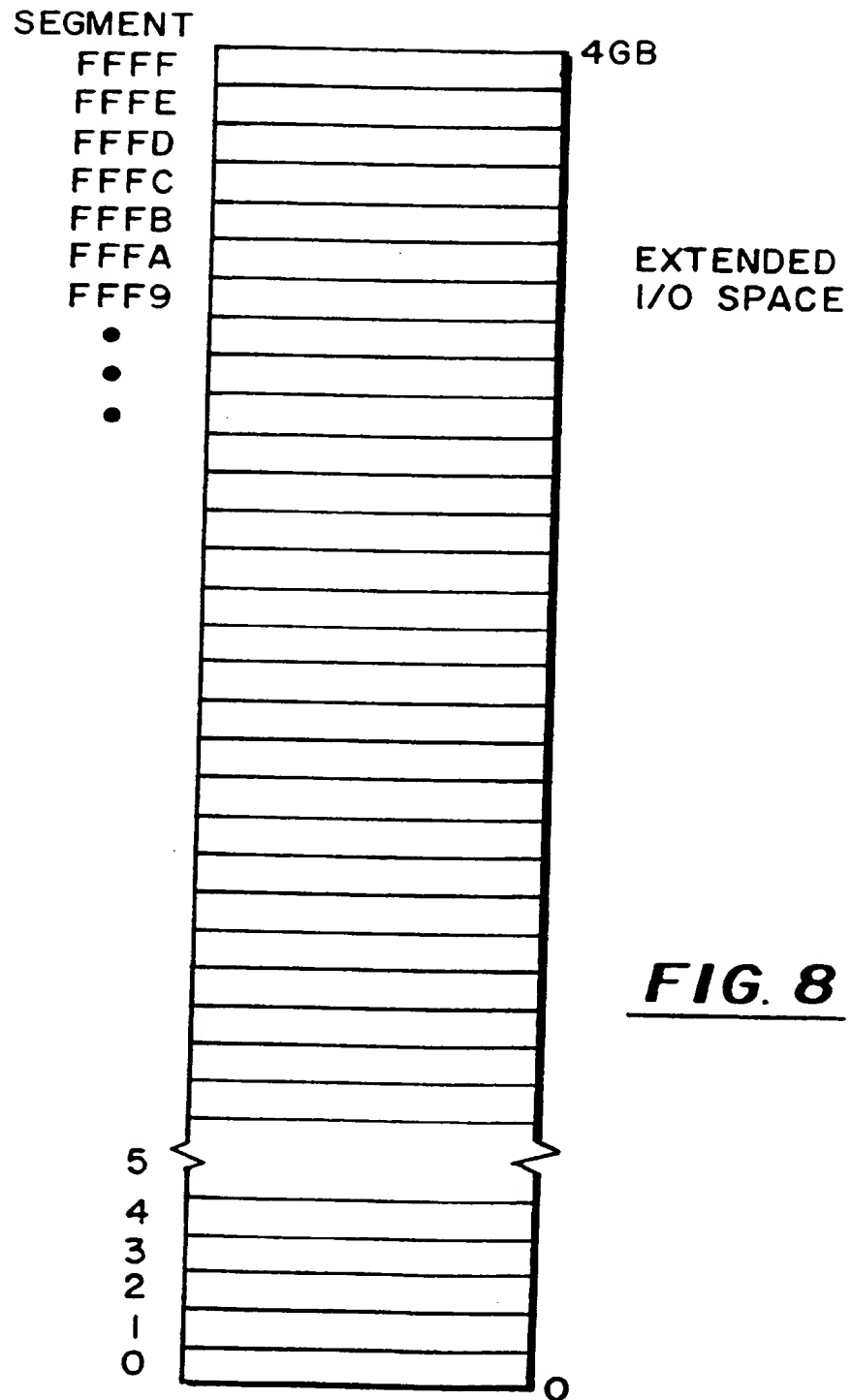
**FIG. 5**

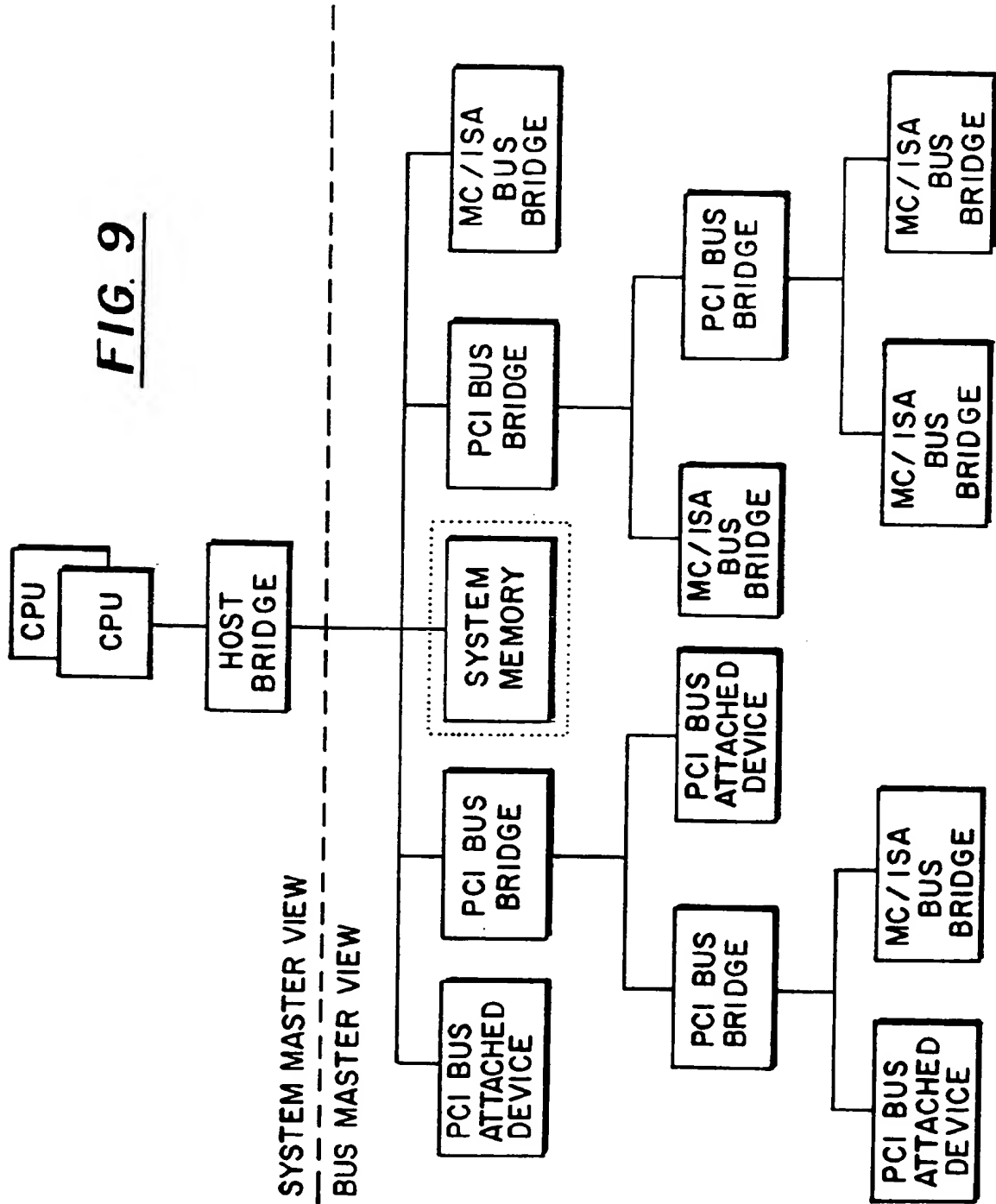


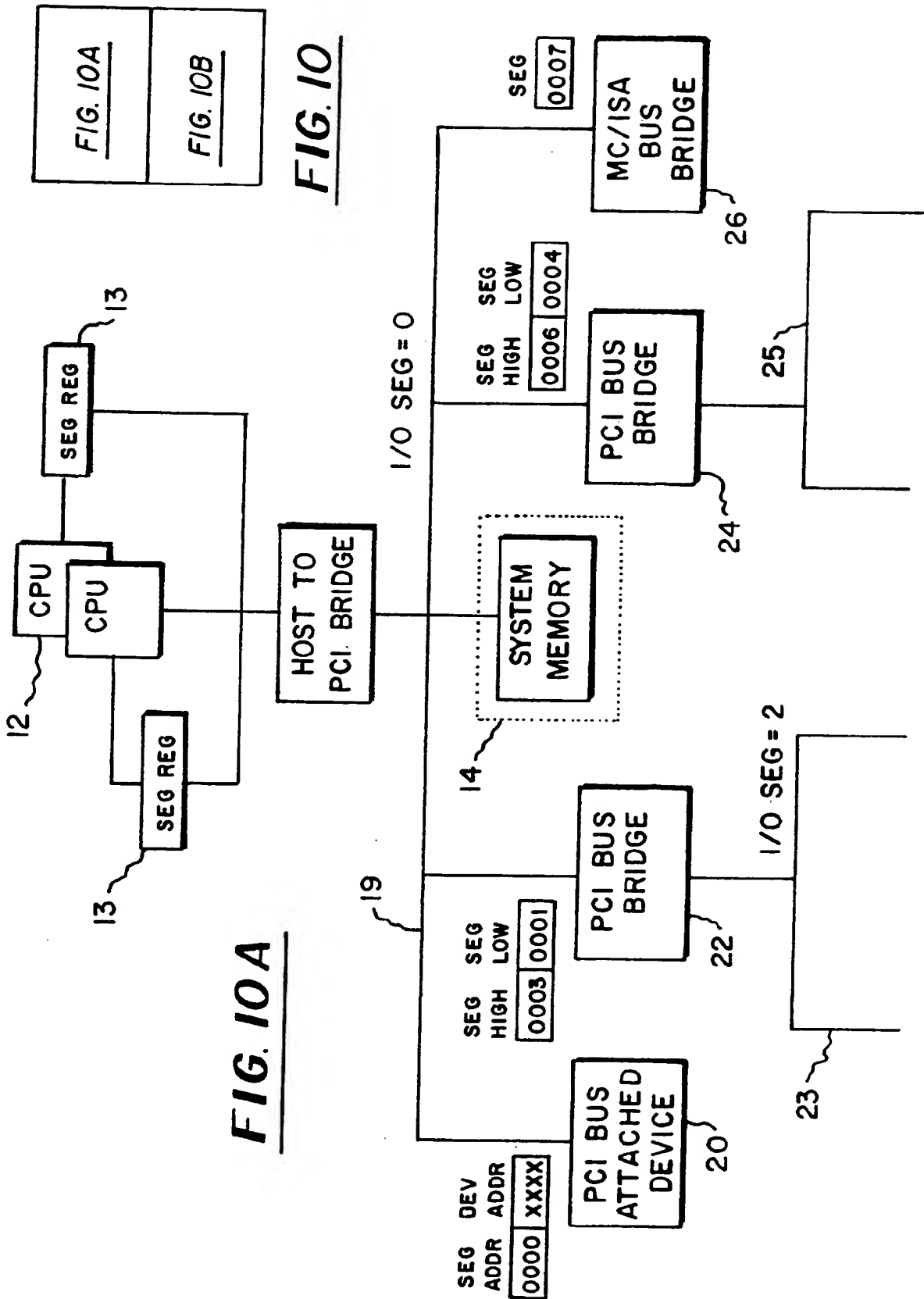
**FIG. 6**

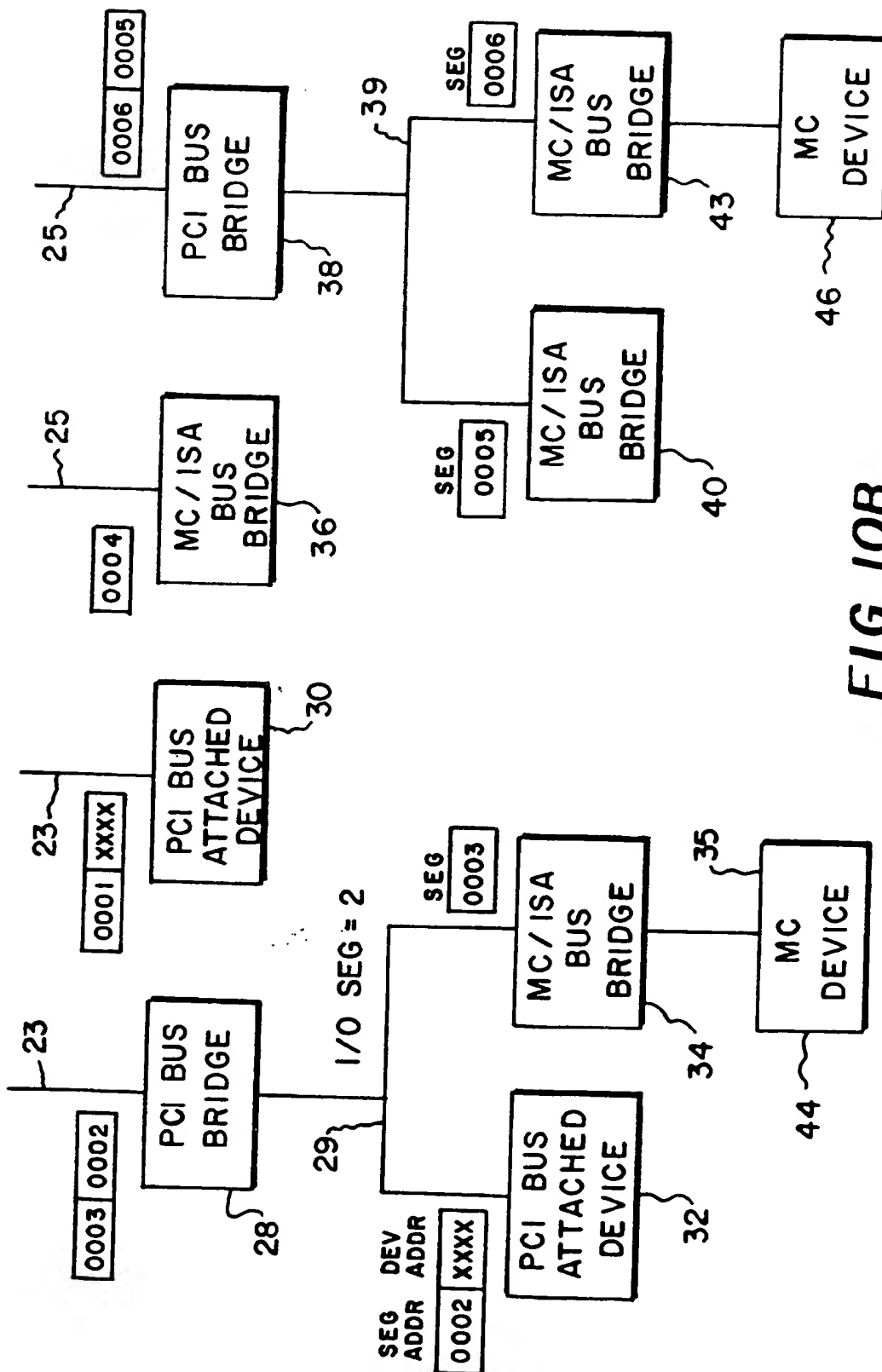




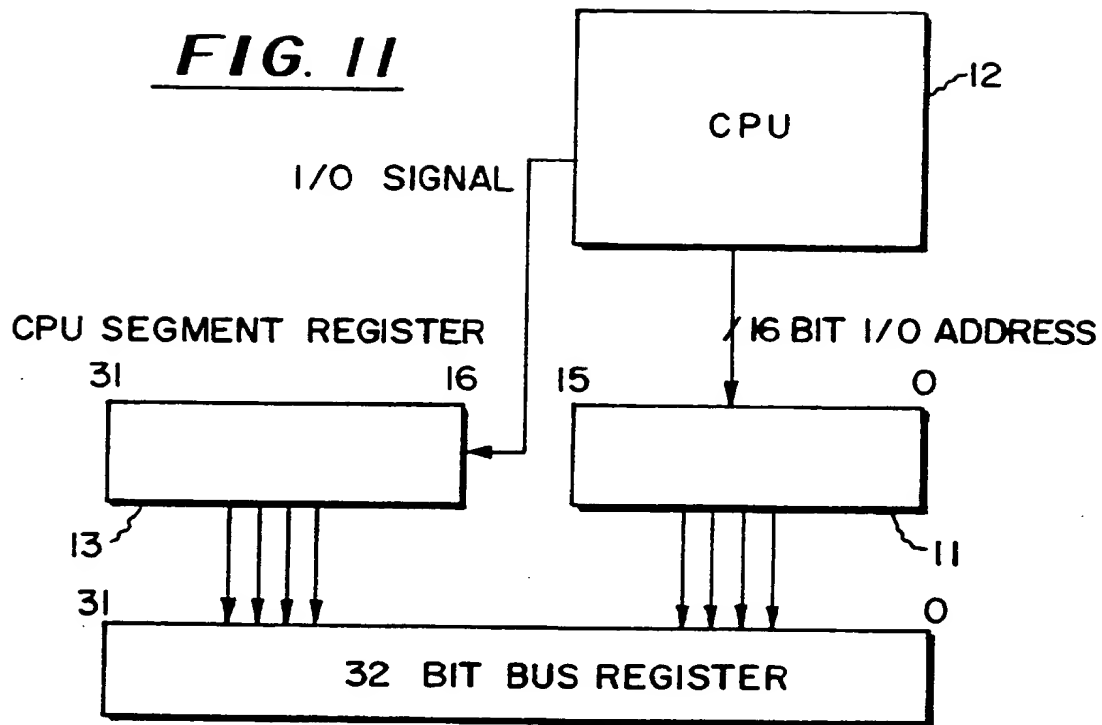


**FIG. 9**





**FIG. 11**



E N B	INPUT SIGNAL	START ADDRESS		END ADDRESS
	OUTPUT SIGNAL	E N B	ADDRESS TRANSLATE	

**FIG. 12**



European Patent  
Office

# EUROPEAN SEARCH REPORT

Application Number  
EP 95 30 3973

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	EP-A-0 362 425 (ADVANCED MICRO DEVICES, INC.)	1-6, 9, 13, 19	G06F13/40
Y	* page 19, line 41 - page 20, line 50; figure 6 *	7, 8, 14-18	
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol.35, no.7, December 1992, NEW YORK US pages 238 - 240 'multiple micro channels' * the whole document *	1-19	
Y	US-A-5 043 877 (BERGER ET AL)	7, 8, 14-18	
A	* column 2, line 60 - column 6, line 39; figure 2 *	1-6, 9-13, 19	
A	NUCLEAR INSTRUMENTS AND METHODS IN PHYSICS RESEARCH A, vol.A292, no.2, 1 July 1990, AMSTERDAM NL pages 424 - 430, XP000137282 D'ANTONE ET AL. 'a DMA link and a Q bus I/O-mapped interface to connect the microvax with the VME bus' * page 428, left column, line 31 - line 45; figure 4 *	1-19	
A	EP-A-0 520 838 (DIGITAL EQUIPMENT CORPORATION) * page 2, column 2, line 35 - page 3, column 4, line 28 *	1-19	TECHNICAL FIELDS SEARCHED (Int.Cl.6)
A	EP-A-0 392 657 (TEKTRONIX, INC.) * page 3, column 4, line 13 - page 4, column 6, line 11 *	1-19	G06F
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 11 August 1995	Examiner Jones, H
<p><b>CATEGORY OF CITED DOCUMENTS</b></p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application I : document cited for other reasons &amp; : member of the same patent family, corresponding document</p>			

EPO FORM 150 (03.92) (P0/C01)

**THIS PAGE BLANK (USPTO)**